AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

LISTING OF CLAIMS:

1. (Currently Amended) A method for in-place updating an old version of a file stored on a storage device to form a new version, the old version including blocks, the method comprising:

determining [[(i)]] a form of said old version, said determined form indicating at which an end of the old version at which free space is located;, and (ii) whether an

comparing a form of a received update package, which indicates a direction in which said received update package updates a file, with said form of said old version to determine if said update package corresponds to is a corresponding update package for said form of said old version; and if so

updating blocks in said old version according to in said direction of said [[corresponding]] update package, if said form of said received update package corresponds to said determined form of said old version, such that giving rise to a new version having an alternative form is generated, [[where]] wherein free space in the new version is located at an [[opposite]] end opposite to that of the old version.

2. (Currently Amended) The method of claim 1, wherein said determined form of said old version is a B-form and wherein said alternative form of said new version is an E-form.

3. (Currently Amended) The method of claim 1, wherein said <u>determined</u> form <u>of said old version</u> is an E-form and wherein said alternative form <u>of said new version</u> is a B-form.

4. (Original) The method of claim 1, further comprising:

for each block in said new version, verifying that content in said new block is successfully stored.

5. (Original) The method of claim 1 wherein said storage device is accommodated in a cellular telephone.

6. (Original) The method of claim 1 further comprising:

determining whether an amount of free space in the old version is too small to allow in-place update of the old version to the new version; and

if so, enlarging said free space to allow said in-place update.

7. (Previously Presented) The method of claim 6, wherein said enlarging includes:

updating said old version to a temporary version having content equivalent to said old version with an alternative form to said old version and a larger free space than in said old version; and

updating said temporary version to form said new version.

8. (Currently Amended) A method for in-place updating an old version of a file stored on a storage device of a remote device to form a new version, the method comprising:

determining a form of said old version indicating ~~at which~~ <u>an</u> end of the old version <u>at which</u> free space is located;

generating an update package<u> based on the end location of free space of the determined form of the old version, such that a direction of the update package for updating said old version</u> ~~that~~ is adapted for said <u>determined</u> form of the old version; and

conveying said update package to said remote device <u>for updating said old version</u>.

9. (Currently Amended) An apparatus for in-place updating an old version of a file stored on a storage device to form a new version, the old version including blocks, the apparatus comprising:

a storage device; and

a processor configured to perform the following:

~~determining (i)~~ <u>determine</u> a form of said old version, <u>said determined form</u>   indicating <u>an</u> ~~at which~~ end of the old version <u>at which</u> free space is located~~;~~ ~~and (ii)~~ <u>compare a form of a received update package, which indicates a direction in which said received update package updates a file, with said form of said old version to determine if said form of said</u> ~~whether an~~ update package <u>corresponds to</u> ~~is a corresponding update package for~~ said <u>determined</u> form <u>of said old version</u>; and ~~if so~~

[[updating]] <u>update</u> blocks in said old version ~~according to said~~ <u>in said direction of said</u> [[corresponding]] update package, <u>if said form of said received</u>

update package corresponds to said determined form of said old version, such that giving rise to a new version having an alternative form is generated, [[where]]

wherein free space in the new version is located at an [[opposite]] end opposite to that of the old version.

10. (Cancelled)

11. (Cancelled)

12. (Currently Amended) A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform a method for in-place updating an old version of a file stored on a storage device to form a new version, the old version including blocks, the method comprising:

determining [[(i)]] a form of said old version, said determined form indicating an at which end of the old version at which free space is located;

comparing a form of a received update package, which indicates a direction in which said received update package updates a file, with said form of said old version to determine if said form of said received, and (ii) whether an update package corresponds to is a corresponding update package for said form of said old version; and if so

updating blocks in said old version in according to said corresponding direction of said received update package, if said form of said received update package corresponds to said form of said old version, such that giving rise to a new version having an alternative form is generated, [[where]] wherein free space in the new version is located at an [[opposite]] end opposite to that of the old version.

13. (Currently Amended) A computer program-product comprising a non-transitory computer useable medium having computer readable program code embodied therein for in-place updating an old version of a file stored on a storage device to form a new version, the old version including blocks, the computer program product comprising:

computer readable program code for causing the computer to determine (i) a form of said old version, said determined form indicating at which end of the old version free space is located, and (ii) whether an if a form of a received update package, indicating a direction in which said received update package updates a file, corresponds to is a corresponding update package for said form of said old version; and

computer readable program code for causing the computer, upon determining the form of said old version and that the received update package corresponds to is a corresponding update package for said form of said old version, to update blocks in said old version in the update direction of according to said corresponding update package, such that giving rise to a new version having an alternative form is generated, [[where]] wherein free space in the new version is located at an [[opposite]] end opposite to that of the old version.

14. (Currently Amended) A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform a method for in-place updating an old version of a file stored on a storage device of a remote device to form a new version, the method comprising:

determining a form of said old version indicating at which an end of the old version at which free space is located;

generating an update package <u>based on the end location of free space of the</u> <u>determined form of the old version, such that a direction of the update package for</u> <u>updating said old version</u> ~~that~~ is adapted for <u>updating</u> said <u>determined</u> form of the old version; and

conveying said <u>generated</u> update package to said remote device <u>for updating</u> <u>said old version</u>.

15. (Currently Amended) A computer program product comprising a non-transitory computer useable medium having computer readable program code embodied therein for in-place updating an old version of a file stored on a storage device of a remote device to form a new version, the computer program product comprising:

computer readable program code for causing the computer to determine a form of said old version<u>, said determined form</u> indicating ~~at which~~ <u>an</u> end of the old version <u>at which</u> free space is located;

computer readable program code for causing the computer to generate an update package <u>based on the end location of free space of the determined form of</u> <u>the old version, such that a direction of the update package for updating said old</u> <u>version</u> [[that]] is adapted for <u>updating</u> said <u>determined</u> form of the old version; and

computer readable program code for causing the computer to convey said <u>generated</u> update package to said remote device <u>for updating said old version</u>.